

EV355228277

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

**APPLICATION FOR LETTERS PATENT**

**Declarative Page View and Click Tracking  
Systems and Methods**

**Inventors:**

Brian S. Christian  
Devdatta Waghdhare

ATTORNEY'S DOCKET NO. MS1-1513US

1 **TECHNICAL FIELD**

2 The systems and methods described herein relate to tracking network client  
3 behavior. More particularly, the systems and methods described herein relate to  
4 page view and click tracking systems and methods that utilize declarative means  
5 initiate tracking.  
6

7 **BACKGROUND**

8 In an effort to provide more useful services, content and user experience on  
9 a network site, such as an Internet site, web page developers study how users make  
10 use of web pages on a particular web site. One way to understand the usefulness  
11 of particular features of a web page are to track what features are frequently  
12 clicked on by users and what features are rarely clicked on by users. In addition,  
13 keeping track of pages and portions of pages viewed by users may also provide  
14 useful information to developers and others.

15 To undertake such study, user patterns are tracked by keeping logs of all the  
16 clicks and page views on a web page. To utilize a click tracking system, each link  
17 contained in a web page has a particular set of click code that is produced based on  
18 the type of the link and the content to which a link references. When a link is  
19 clicked, a client (i.e., user computer) is first directed to a tracking server where the  
20 click information gets logged. Thereafter, the client is redirected to a destination  
21 specified in the link. User patterns are derived from statistics generated from log  
22 files on the tracking server.

23 Each link has an associated script (typically JavaScript) function that details  
24 the click code, destination address, etc. Each link script may also have parameters  
25 to define if the link is a link to a page on an internal site or an external site. Each

1 link also may be programmed to behave differently depending on if the link is a  
2 link to a web page or a link to media content. Furthermore, the link script may  
3 depend on various external files for redirection code.

4 All this requires a significant amount of code to be manually included in  
5 every file to generate an appropriate click tracking element for a particular click  
6 tracking system. This also tends to create an undesirable amount of external  
7 dependencies. Unfortunately, the tedious nature of providing unique click code for  
8 each link in a web page necessarily creates a lot of room for errors by page  
9 developers. To avoid these problems, a standardized, declarative solution is  
10 desirable that would provide a more convenient and efficient way for page  
11 developers to provide certain functionality (such as click tracking and/or page  
12 view tracking) to their web pages.

## 13 14 **SUMMARY**

15 Declarative page view and click tracking systems and methods are  
16 described. The present description is related to a click tracking/page view tracking  
17 system, although the described techniques may be used in other systems for other  
18 purposes. To initiate click tracking within a web page, a page developer must only  
19 include a link identifier in a parameterized link, which is HTML (HyperText  
20 Markup Language) compliant. One or more custom function calls (or build  
21 anchor tag calls) are defined herein that can be used inside of a link element or an  
22 anchor tag to have the link participate in a tracking system. As a result, page  
23 developers do not have to customize click code for each unique link.

24 When a selectable link is clicked on by a client, the link identifier triggers a  
25 click tracking system to initiate click tracking for that particular link. The click

1 tracking system replaces the link with a modified link automatically built to  
2 system specifications. The click on the link is then logged by a tracking server  
3 and the client is redirected to a destination specified in the link. If the click cannot  
4 be logged within a specified timeout period, then the link is redirected to its  
5 destination without logging the click to avoid an unsatisfactory user experience.

6 The techniques described herein are “declarative” in that the functional  
7 aspects of the techniques are performed outside individual page code, and that  
8 there is only a single function call for all of the different kinds of links that may be  
9 used on a web page. External dependencies and individual code related to each  
10 link contained in the web page are eliminated. In addition, the code required to  
11 implement click tracking and handling is transparent to a page developer. This  
12 significantly eases the burden of providing certain functionality (such as click  
13 tracking) to each link that is included in a web page.

#### 14 15 **BRIEF DESCRIPTION OF THE DRAWINGS**

16 The same numbers are used throughout the document to reference like  
17 components and/or features.

18 Fig. 1 is a block diagram illustrating an exemplary network environment.

19 Fig. 2 is a representation of a selectable link that may be included in a web  
20 page.

21 Fig. 3 is a representation of a tracking identifier as described herein.

22 Fig. 4 is a representation of a tracking system modified selectable link as  
23 described herein.

24 Fig. 5 is a flow diagram depicting a methodological implementation of a  
25 page view click tracking system.

Fig. 6 illustrates a general computer environment, which can be used to implement the techniques described herein.

#### **DETAILED DESCRIPTION**

The following depictions describe one or more exemplary systems and/or methods for declarative page view and click tracking. The examples described are but a few examples of various manners in which the subject matter of the appended claims may be implemented. The described examples are not intended to limit the scope of the appended claims in any manner, but are shown to accurately describe the best mode of carrying out the invention delineated by the claims.

The examples described below relate to ASP.NET technology, but may be implemented in one or more other types of web services framework. ASP (Active Server Pages) is a server-side scripting technique promulgated by MICROSOFT CORP® that enables server execution of scripts embedded in web pages. ASP is included in the WINDOWS® family of operating systems.

An ASP file (a file having an .asp extension) may contain HTML, text, XML and/or one or more scripts. Scripts in the ASP file are executed by a server. When a client web browser requests an HTML file from a server, the server merely returns the requested HTML file. When a client web browser requests an ASP file from a server, an Internet Information Server (IIS) in the server passes the request to an ASP module. An ASP engine reads the file, line by line, and executes the scripts in the file. The ASP file is then returned to the client web browser as a plain HTML file.

1 Although the present examples will focus on ASP technology, it is noted  
2 that the examples may be implemented with any other form of web services  
3 scripting technology without departing from the scope of the claimed systems and  
4 methods.

5 The systems and methods described below relate to a web page  
6 development framework within which page developers can utilize a parameterized  
7 link element to provide click tracking for a link on a web page instead of  
8 associating unique code with each link that executes upon selection of the link by  
9 a user. The information collected and transmitted to an external system (such as,  
10 in the present case, a tracking server) is provided in parameters associated with the  
11 element. The element parameters are used to automatically generate a URL  
12 (Universal Resource Locator) that is properly constructed to provide an advanced  
13 click tracking function with the link.

#### 14 15 **Exemplary Network Environment**

16 Fig. 1 is a block diagram that illustrates an exemplary network environment  
17 100. The exemplary network environment 100 includes a web server 102 that  
18 communicates over the Internet 104 with a tracking server 106 and a client  
19 computer 108 (hereinafter "client 108"). The web server 102 provides web  
20 content to the client 108 as well as multiple other clients (not shown).

21 The web server 102 includes a tracking module 110 and several web pages  
22 112(1) - 112(n). The tracking module 110 includes client code 114 that is injected  
23 into the web pages 112 under certain conditions. For exemplary purposes, each  
24 web page 112 is shown having the client code 114 included therein. However, as  
25 will be described in greater detail below, the client code 114 is not necessarily

1 originally included in the web page 112 but may have the client code 114 injected  
2 into it by the tracking module 110. Each particular web page 112(1) - 112(n)  
3 includes at least one selectable link 116(1) - 116(n), respectively.

4 The tracking server 106 includes a click tracking application 120 that is  
5 configured to track page clicks and, in some implementations, page views. The  
6 tracking server 106 also includes a log files database 122 that stores information  
7 obtained by the click tracking application 120.

8 The client 108 includes a resource file 118 that stores script functions used  
9 in a tracking process. During tracking operations, the resource file 118 contents  
10 are stored on the client 108 by the tracking module 110. The client 108 may also  
11 include the client code 114 if an implementation is such that the client code 114 is  
12 downloaded to the client 108 by the web server 102. However, the  
13 implementations described herein do not entail downloading the client code 114 to  
14 the client 108.

15 The client 108 communicates with a display device 130 that has the ability  
16 to display one or more of the web pages 112 (or other web pages, not shown). The  
17 selectable link 116 of a particular web page 112 is selectable on the display  
18 device.

19 The functions of the elements shown in Fig. 1 will be discussed in greater  
20 detail below, with respect to the following figures.

### 21 22 **Exemplary Selectable Link**

23 Fig. 2 is an illustration of a selectable link 200 - similar to the selectable  
24 links 116 shown in Fig. 1 - that may be included in a web page. The selectable  
25 link 200 is shown in its original unmodified form, similar to what would be

1 included in a web page by a web developer. The selectable link 200 is an HTML  
2 compliant link element.

3 The selectable link 200 includes a label 201 and a URL 202 that a client  
4 computer is directed to when a client computer user selects the selectable link 200.  
5 The label 201 may be a function call or a build anchor tag call name that initiates  
6 the link. For convenience, any further reference to a function call may also denote  
7 a build anchor tag call. The selectable link 200 also includes one or more  
8 parameters 204(1) - 204(n). The URL 202 and the one or more parameters 204 are  
9 typical of those included in an HTML-compliant link.

10 An example of the label 201 that is utilized in MICROSOFT®  
11 WINDOWS® family of operating systems is **GetHrefText** (in the case of a link  
12 function call), or **BuildAnchorTag** (in the case of a build anchor tag call). These  
13 labels are exemplary only and may vary in different systems and/or  
14 implementations.

15 Examples of the parameters 204 can include *internallink*, *stream*, *linkid*,  
16 *objectid*, *windowparams* and *onclickscript*. Therefore, a function call may be  
17 represented as:

18 <a **GetHrefText**(url, internallink, stream, linkid, objectid, onclickscript)>link </a>

19 Note that the windowparams parameter may also be included in the event  
20 that a link opens a popup window.

21 Similarly, a build anchor tag call in a WINDOWS®-based system may be  
22 represented as:

23 **BuildAnchorTag**(url, internallink, stream, linkid, objectid, onclickscript, linktext).



1 It is also noted that the functionality described above may also be  
2 implemented inside ASP.NET user controls. Therefore, developers don't have to  
3 implement the Page Base Class inside custom user controls. The syntax for doing  
4 so is:

5 **public static string GetHrefText(...)** and  
6 **public static string BuildAnchorTag(...).**

7 The selectable link 200 also includes a link identifier 206. The link  
8 identifier 206 is assigned by a web page developer who develops the web page  
9 that includes the selectable link 200. As will be discussed in greater detail below,  
10 the link identifier 206 may be maintained or modified by a click tracking system.

11 Web page developers follow a fixed pattern to develop the link identifier  
12 206 for a web page that runs under particular system. The fixed pattern may differ  
13 between different systems. The web page developer derives the link identifier 206  
14 by following the fixed pattern, then includes the link identifier 206 in a link of a  
15 web page. If the fixed pattern is followed to derive the link identifier 206, then the  
16 link identifier 206 will probably remain the same when the link is modified to  
17 include a tracking identifier (see below). If, however, the link identifier 206 does  
18 not conform to system specifications, the system may change the link identifier  
19 206 during tracking processing.

20 In at least one implementation, the link identifier 206 is divided into two  
21 parts, to-wit: an area identifier 208 (i.e., the section of a web page where the link is  
22 located), and a link type identifier 210. For example, if the link is inside a third  
23 feature on the web page, the area identifier 208 is F2. If the link is a link on a first  
24 sub-feature, then the area identifier 208 is S0.

1       The link type identifier 210 is the type of link followed by a count of that  
2 link inside the area in which it is located. For example, if the link is a third text  
3 link, the link type identifier 210 is T2. If the link is a first graphic link in a  
4 particular area, the link type identifier 210 is G0.

5       The area identifier 208 and the link type identifier 210 are combined to  
6 form the link identifier 206. In the above examples, the link identifiers 206 that  
7 would result from the exemplary area identifiers 208 and link type identifiers 210  
8 are F2T1 and S0G0, respectively.

9       As will be discussed below, the link identifier 204 identifies the selectable  
10 link 200 to a click tracking system (tracking module 110, Fig. 1) as a link that  
11 should be tracked. Inclusion of the link identifier 204 in the selectable link 200  
12 does not affect the HTML-compliant status of the selectable link 200.

13       The inclusion of the link identifier 200 is the declaration made in the web  
14 page that keeps the selectable link 200 from having to have unique click code  
15 associated with it. The tracking module 110 identifies the selectable link 200 as  
16 one that should be tracked and - by virtue of the inclusion of the link identifier 204  
17 - performs click tracking on the selectable link 200 if and when the link 200 is  
18 selected.

### 19       **Exemplary Tracking Identifier**

20       Fig. 3 illustrates an exemplary tracking identifier 300 that is substituted into  
21 the selectable link (200, Fig. 2) by a tracking system (e.g. tracking module 110,  
22 Fig. 1). The tracking identifier 300 includes a container identifier 302 that  
23 uniquely identifies a container (not shown) that contains the selectable link 200  
24 (for instance, if the selectable link 200 is a link inside a specific group of links).  
25

1       The tracking identifier 300 also includes a link index 304 that is an index of  
2       the selectable link 200 inside the container that is identified by the container  
3       identifier 302. The link index 304 denotes a link's position relative to other links  
4       within the same container. For example, if the selectable link is the second link in  
5       the specified container, the selectable link 200 may have a link index of "2", "L2"  
6       or the like, depending on the particular implementation.

7       The tracking identifier 300 also includes a link identifier 306 that may or  
8       may not be the same as the link identifier 206 (Fig. 2) that is assigned by the page  
9       developer. The particular implementation of the tracking module 110 determines  
10      the derivation of the link identifier 306. If a particular implementation of tracking  
11      module 110 does not utilize the original link identifier 206, then the original link  
12      identifier 206 is replaced with a modified link identifier 306 generated by the  
13      tracking module 110.

#### 14      **Exemplary Modified Selectable Link**

15      Fig. 4 illustrates an exemplary selectable link 400 that has been modified  
16      by the tracking module 110, Fig. 1. In the present example, the selectable link 400  
17      shown in Fig. 4 is the same as the selectable link 200 shown in Fig. 2, except that  
18      the link identifier 206 included in the selectable link 200 of Fig. 2 has been  
19      replaced in Fig. 4 by a tracking identifier 406 that is similar to the tracking  
20      identifier 306 shown and described in Fig. 3. The modified selectable link 400  
21      also includes a URL 402 and one or more parameters 404(1) - 404(n).

22      When the tracking application 110 identifies the selectable link 200 as  
23      being configured to invoke click tracking (or page view tracking, or both), then the  
24      tracking application 110 removes the selectable link 200, transforms the selectable  
25      link 200 into the modified selectable link 400 and inserts the modified selectable

1 link 400 into the web page. As will be discussed in greater detail, below, the  
2 tracking module 110 then logs the selection of the link 200/400 and redirects the  
3 selector to a web page identified by the URL 202/402 included in the link  
4 200/400.

### 6 **Exemplary Methodological Implementation**

7 Fig. 5 is a flow diagram 500 depicting a methodological implementation of  
8 a page view click tracking system. The flow diagram 500 depicts activities that  
9 occur in the web server 102 and the client 108 and is divided accordingly. In the  
10 following discussion, continuing reference will be made to previous figures and  
11 the elements contained therein.

12 At block 502, the client computer 108 requests a web page 112(1) from the  
13 web server 102 by clicking on the selectable link 116(1) of the web page 112(1).  
14 The web server 102 - more particularly, the tracking module 110 - accesses the  
15 web page 112(1) and inspects the selectable link 116(1), and any others that may  
16 be included in the web page 112(1), at block 504.

17 At this point, the tracking module 110 may also be configured to inspect the  
18 client 108 to determine if the client 108 is of a configuration that is desirable to  
19 track. The parameters for this determination may be configured for each particular  
20 implementation of the tracking module 110. For example, the tracking module  
21 110 may be configured to track clicks and page views only for client computers  
22 that run a certain version of web browser, such as INTERNET EXPLORER® 5.0  
23 or higher. Or tracking may only be desirable for client computers having certain  
24 hardware configurations.

1        If the tracking module 110 determines that the client 108 is not one that is  
2 desirable to track ("No" branch, block 506), then the process terminates. If the  
3 tracking module 110 determines that the client 108 is suitable for tracking ("Yes"  
4 branch, block 506), then the tracking module 110 replaces the selectable link  
5 116(1) with a modified selectable link (similar to the modified selectable link 400  
6 shown in Fig. 4) at block 508.

7        Replacing the original selectable link 116(1) provides the web server 102  
8 with an opportunity to place the link into an appropriate condition for use with the  
9 tracking application 110. For instance, the tracking identifier (300, Fig. 3) is  
10 added to the selectable link 116(1) so that the tracking module 110 and/or the click  
11 tracking application 120 can properly record the selection of the selectable link  
12 116(1). Also, if the selectable link 116(1) contains some minor errors, the tracking  
13 application 110 may be able to remedy those errors at this point. Such errors may  
14 include an incorrect URL designation or the like.

15        At block 510, the tracking application 110 injects the client code 114 into  
16 the web page 112(1). By injecting the client code 114 into the web page 112(1),  
17 the tracking module 110 provides the client 108 with code that may be required to  
18 properly work with the click tracking application 120. In one or more alternative  
19 implementations, the client code 114 could be downloaded to the client 108, but  
20 this may not be desirable in many cases.

21        The configuration of the client code 114 depends on a client-side scripting.  
22 The client code 114 depends on different script functions which are stored in the  
23 resource file 118 on the client 108 so that a page developer does not need to  
24 include the script files in every project.  
25

1 Based on values included in the link parameters, the tracking module 110  
2 emits appropriate script code to the client 108 which is used by the click tracking  
3 application 120.

4 In one particular implementation (WINDOWS®), a value "wmis.page.track  
5 = 42" causes javascript functions supporting click tracking to be emitted. A value  
6 "wmis.page.track = 42" emits code to redirect to an external partner and to  
7 simulate an external click so that external partners can identify the entity that  
8 referred the user to the partner site. A value "wmis.page.clickback = 42" emits a  
9 function that completes the link after receiving a response from the tracking server  
10 106.

11 It is noted that the above examples are configured for use with  
12 WINDOWS® compatible applications. Other client code may be used with the  
13 present technique with other application, programs, systems, etc. as may be  
14 desired for any particular implementation.

15 At block 512, the requested web page 112(1) is loaded for the client 108.  
16 The tracking module 110 confirms that the web page 112(1) properly loads and is  
17 viewable at the client 108 and, if so, sends a page view event to the tracking server  
18 106 where the click tracking application 120 records the event in the log files 122.

19 When the selectable link 116(1) is clicked at the client 108 (block 518), the  
20 client code 114 begins execution at block 520. At this point, the client code 114  
21 could send a click tracking event to the tracking server 106. However, in the  
22 described implementation, the client code 114 first determines the viability of the  
23 tracking server 106 at block 522. This may be accomplished in one of various  
24 implementations.  
25

1 In at least one implementation, the client code initiates an HTTP  
2 (HyperText Transfer Protocol) request to occur, namely, a blank page to load at  
3 the tracking server 106. If the page load is confirmed within a predefined timeout  
4 period ("Yes" branch, block 522), then the response is timely and a click tracking  
5 event is sent to the tracking server 106 at block 524.

6 If the page load cannot be confirmed within the timeout period ("No"  
7 branch, block 522), then the tracking procedure is skipped. This way, a client user  
8 is not inconvenienced by having to wait a significant amount of time for the link to  
9 occur. Also, it helps to prevent a denial of service attack on the tracking server  
10 106 that may be caused by submitting numerous requests to the tracking server  
11 106 that are never timed out.

12 In at least one implementation, the predefined timeout period is one and  
13 one-half (1½) seconds. However, the timeout period may be configured as desired  
14 in any particular implementation.

15 The client 108 is then linked to a web page identified as the destination in  
16 the selectable link 116(1) at block 526 and the client 108 can view the web page  
17 identified as the destination in the selectable link 116(1) at block 528.

### 18 19 **Exemplary Operating Environment**

20 Fig. 6 illustrates a general computer environment 600, which can be used to  
21 implement the techniques described herein. The computer environment 600 is  
22 only one example of a computing environment and is not intended to suggest any  
23 limitation as to the scope of use or functionality of the computer and network  
24 architectures. Neither should the computer environment 600 be interpreted as  
25

1 having any dependency or requirement relating to any one or combination of  
2 components illustrated in the exemplary computer environment 600.

3 Computer environment 600 includes a general-purpose computing device in  
4 the form of a computer 602. Computer 602 can be, for example, a client 108 or  
5 server 102 of Fig. 1. The components of computer 602 can include, but are not  
6 limited to, one or more processors or processing units 604, a system memory 606,  
7 and a system bus 608 that couples various system components including the  
8 processor 604 to the system memory 606.

9 The system bus 608 represents one or more of any of several types of bus  
10 structures, including a memory bus or memory controller, a peripheral bus, an  
11 accelerated graphics port, and a processor or local bus using any of a variety of  
12 bus architectures. By way of example, such architectures can include an Industry  
13 Standard Architecture (ISA) bus, a Micro Channel Architecture (MCA) bus, an  
14 Enhanced ISA (EISA) bus, a Video Electronics Standards Association (VESA)  
15 local bus, and a Peripheral Component Interconnects (PCI) bus also known as a  
16 Mezzanine bus.

17 Computer 602 typically includes a variety of computer readable media.  
18 Such media can be any available media that is accessible by computer 602 and  
19 includes both volatile and non-volatile media, removable and non-removable  
20 media.

21 The system memory 606 includes computer readable media in the form of  
22 volatile memory, such as random access memory (RAM) 610, and/or non-volatile  
23 memory, such as read only memory (ROM) 612. A basic input/output system  
24 (BIOS) 614, containing the basic routines that help to transfer information  
25 between elements within computer 602, such as during start-up, is stored in ROM



1 612. RAM 610 typically contains data and/or program modules that are  
2 immediately accessible to and/or presently operated on by the processing unit 604.

3 Computer 602 may also include other removable/non-removable,  
4 volatile/non-volatile computer storage media. By way of example, Fig. 6  
5 illustrates a hard disk drive 616 for reading from and writing to a non-removable,  
6 non-volatile magnetic media (not shown), a magnetic disk drive 618 for reading  
7 from and writing to a removable, non-volatile magnetic disk 620 (e.g., a "floppy  
8 disk"), and an optical disk drive 622 for reading from and/or writing to a  
9 removable, non-volatile optical disk 624 such as a CD-ROM, DVD-ROM, or other  
10 optical media. The hard disk drive 616, magnetic disk drive 618, and optical disk  
11 drive 622 are each connected to the system bus 608 by one or more data media  
12 interfaces 625. Alternatively, the hard disk drive 616, magnetic disk drive 618,  
13 and optical disk drive 622 can be connected to the system bus 608 by one or more  
14 interfaces (not shown).

15 The disk drives and their associated computer-readable media provide non-  
16 volatile storage of computer readable instructions, data structures, program  
17 modules, and other data for computer 602. Although the example illustrates a hard  
18 disk 616, a removable magnetic disk 660, and a removable optical disk 624, it is to  
19 be appreciated that other types of computer readable media which can store data  
20 that is accessible by a computer, such as magnetic cassettes or other magnetic  
21 storage devices, flash memory cards, CD-ROM, digital versatile disks (DVD) or  
22 other optical storage, random access memories (RAM), read only memories  
23 (ROM), electrically erasable programmable read-only memory (EEPROM), and  
24 the like, can also be utilized to implement the exemplary computing system and  
25 environment.

1 Any number of program modules can be stored on the hard disk 616,  
2 magnetic disk 620, optical disk 624, ROM 612, and/or RAM 610, including by  
3 way of example, an operating system 626, one or more application programs 628,  
4 other program modules 630, and program data 632. Each of such operating  
5 system 626, one or more application programs 628, other program modules 630,  
6 and program data 632 (or some combination thereof) may implement all or part of  
7 the resident components that support the distributed file system.

8 A user can enter commands and information into computer 602 via input  
9 devices such as a keyboard 634 and a pointing device 636 (e.g., a "mouse").  
10 Other input devices 638 (not shown specifically) may include a microphone,  
11 joystick, game pad, satellite dish, serial port, scanner, and/or the like. These and  
12 other input devices are connected to the processing unit 604 via input/output  
13 interfaces 640 that are coupled to the system bus 608, but may be connected by  
14 other interface and bus structures, such as a parallel port, game port, or a universal  
15 serial bus (USB).

16 A monitor 642 or other type of display device can also be connected to the  
17 system bus 608 via an interface, such as a video adapter 644. In addition to the  
18 monitor 642, other output peripheral devices can include components such as  
19 speakers (not shown) and a printer 646 which can be connected to computer 602  
20 via the input/output interfaces 640.

21 Computer 602 can operate in a networked environment using logical  
22 connections to one or more remote computers, such as a remote computing device  
23 648. By way of example, the remote computing device 648 can be a personal  
24 computer, portable computer, a server, a router, a network computer, a peer device  
25 or other common network node, and the like. The remote computing device 648 is

1 illustrated as a portable computer that can include many or all of the elements and  
2 features described herein relative to computer 602.

3 Logical connections between computer 602 and the remote computer 648  
4 are depicted as a local area network (LAN) 650 and a general wide area network  
5 (WAN) 652. Such networking environments are commonplace in offices,  
6 enterprise-wide computer networks, intranets, and the Internet.

7 When implemented in a LAN networking environment, the computer 602 is  
8 connected to a local network 650 via a network interface or adapter 654. When  
9 implemented in a WAN networking environment, the computer 602 typically  
10 includes a modem 656 or other means for establishing communications over the  
11 wide network 652. The modem 656, which can be internal or external to computer  
12 602, can be connected to the system bus 608 via the input/output interfaces 640 or  
13 other appropriate mechanisms. It is to be appreciated that the illustrated network  
14 connections are exemplary and that other means of establishing communication  
15 link(s) between the computers 602 and 648 can be employed.

16 In a networked environment, such as that illustrated with computing  
17 environment 600, program modules depicted relative to the computer 602, or  
18 portions thereof, may be stored in a remote memory storage device. By way of  
19 example, remote application programs 658 reside on a memory device of remote  
20 computer 648. For purposes of illustration, application programs and other  
21 executable program components such as the operating system are illustrated herein  
22 as discrete blocks, although it is recognized that such programs and components  
23 reside at various times in different storage components of the computing device  
24 602, and are executed by the data processor(s) of the computer.

1 Various modules and techniques may be described herein in the general  
2 context of computer-executable instructions, such as program modules, executed  
3 by one or more computers or other devices. Generally, program modules include  
4 routines, programs, objects, components, data structures, etc. that perform  
5 particular tasks or implement particular abstract data types. Typically, the  
6 functionality of the program modules may be combined or distributed as desired in  
7 various embodiments.

8 An implementation of these modules and techniques may be stored on or  
9 transmitted across some form of computer readable media. Computer readable  
10 media can be any available media that can be accessed by a computer. By way of  
11 example, and not limitation, computer readable media may comprise "computer  
12 storage media" and "communications media."

13 "Computer storage media" includes volatile and non-volatile, removable  
14 and non-removable media implemented in any method or technology for storage  
15 of information such as computer readable instructions, data structures, program  
16 modules, or other data. Computer storage media includes, but is not limited to,  
17 RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM,  
18 digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic  
19 tape, magnetic disk storage or other magnetic storage devices, or any other  
20 medium which can be used to store the desired information and which can be  
21 accessed by a computer.

22 "Communication media" typically embodies computer readable  
23 instructions, data structures, program modules, or other data in a modulated data  
24 signal, such as carrier wave or other transport mechanism. Communication media  
25 also includes any information delivery media. The term "modulated data signal"

1 means a signal that has one or more of its characteristics set or changed in such a  
2 manner as to encode information in the signal. By way of example, and not  
3 limitation, communication media includes wired media such as a wired network or  
4 direct-wired connection, and wireless media such as acoustic, RF, infrared, and  
5 other wireless media. Combinations of any of the above are also included within  
6 the scope of computer readable media.

### 7 8 **Conclusion**

9 The systems and methods described herein provide a convenient way to  
10 incorporate click track and/or page view tracking into a web page by declarative  
11 means. As such, web page developers can utilize desirable functionality without  
12 having to provide unique click code for each link in a web page. This results in  
13 more uniform programming and reduces errors in page links.

14 Although the description above uses language that is specific to structural  
15 features and/or methodological acts, it is to be understood that the invention  
16 defined in the appended claims is not limited to the specific features or acts  
17 described. Rather, the specific features and acts are disclosed as exemplary forms  
18 of implementing the invention.